

بسمه تعالی



گزارش فاز سوم پروژه

درس معماری کامپیوتر

استاد: دکتر سربازی آزاد

اعضای گروه:

پویا یوسفی (۹۸۱۷۱۲۲۳)

سجاد پاک‌سیما (۹۸۱۰۶۲۸۶)

محمد رضا احمدی تشنیزی (۹۸۱۷۰۶۴۶)

مهدیه ابراهیم‌پور (۹۸۱۷۰۶۲۴)

نیمسال بهار ۱۴۰۱

مقدمه

در این فاز، دستورالعمل‌ها به صورت خط لوله اجرا می‌شوند؛ بطوریکه هر استیج از خط لوله در یک کلاک با پریود مناسب اجرا می‌شود. این کار باعث می‌شود تا دستورالعمل‌های بیشتری در یک بازه زمانی مشخص اجرا شوند.

شرح کار

استیج‌ها

در پردازنده میپس که یک پردازنده RISC است، ما از ۵ استیج زیر استفاده کردیم:

- IF (Instruction Fetch): در این استیج مقدار PC آپدیت شده و مقدار خروجی به Instruction Memory داده می‌شود تا دستورالعمل جدید fetch شود.
- ID (Instruction Decode): در این استیج، دستورالعملی که از استیج قبل بافر شده‌است، دیکود می‌شود و سیگنال‌های کنترلی تولید می‌شوند. همچنین واحد مقایسه در استیج قرار داده شده تا از control hazard تولیدی ناشی از jump, branch و jr جلوگیری شود. (Early Branch) در نهایت مقدار آدرس دستورالعمل بعدی، تولید و به PC پاس داده می‌شود. اگر یکی از دستورات فوق وجود داشته باشد، سیگنالی به نام program_flow_change_taken فعال شده و بافر IF/ID را فلاش می‌کند تا دیتای نادرست تولید نشود. سپس دستور بعد توسط PC_EXE مشخص می‌شود.
- EXE (Instruction Execute): در این استیج، ALU قرار داده شده‌است تا عملیات محاسباتی را انجام دهد. در ورودی ALU ها و همچنین مقداری که قرار است در حافظه ذخیره شود، یک Mux با سه ورودی قرار داده شده است تا عملیات Forwarding را انجام دهد؛ در واقع بسته به اینکه رجیستر فعلی با رجیستر استیج‌های بعدی برابر باشد، ورودی درست select می‌شود. عملیات select توسط واحدی به نام Forwarding Unit انجام می‌شود.
- MEM (Memory Access): در این استیج، عملیات load و store از حافظه انجام می‌شود. به این منظور، یک cache قرار داده شده است که مطابق با فاز دو می‌باشد. در مواقعی که نیاز باشد، cache با حافظه ارتباط می‌گیرد.
- WB (Write Back): در این استیج، رجیستر مقصد و مقداری که قرار است داخلش نوشته شود، مشخص شده و به واحد رجیستر فایل ارسال می‌شود.

توجه شود که همه استیج‌ها و بافرها در یک کلاک مشترک اجرا می‌شوند. همان طور که مشاهده می‌کنید، استیج اول وظیفه دریافت دستورالعمل، استیج دوم وظیفه دیکود دستور و مشخص کردن سیگنال‌های کنترلی، استیج سوم وظیفه محاسبات، استیج چهارم وظیفه ارتباط با حافظه و استیج پنجم وظیفه مشخص کردن مقدار ذخیره شده در رجیستر را به عهده دارد. دلیل انتخاب این پنج استیج این بود که هرکدام کار مستقلی را انجام می‌دهند و در یک کلاک با پریود مناسب اجرا می‌شوند.

بافرها

بین هر دو استیج، یک بافر در نظر گرفتیم که با کلاک کار می‌کند و مقادیر لازم برای استیج‌های بعدی را بافر می‌کند.

Forwarding Unit

این واحد برای کاهش میزان data hazard اضافه شده است؛ زیرا ممکن است دو دستورالعمل متوالی به هم وابسته باشند و دیتای نادرست پردازش شود. برای رجیسترهایی که ورودی ALU هستند و همچنین رجیستری که قرار است در حافظه ذخیره

شود، Mux با سه ورودی قرار داده شده که select آنها براساس این واحد انجام می‌شود. سه ورودی Mux ها شامل رجیستری که از استیج قبل بافر شده، مقدار خروجی alu_result در استیج MEM و مقدار rd_data در استیج WB است. اگر رجیستری که قرار است پردازش شود یا رجیستری که در مرحله MEM قرار دارد و پردازش شده برابر باشد، مقدار select برابر ۱ و اگر با رجیستری که در مرحله WB قرار دارد و پردازش شده برابر باشد، مقدار آن را ۲ قرار می‌دهیم.

Hazard Detection Unit

این واحد برای تشخیص hazard موجود در دستورالعمل‌ها قرار داده شده است. زمانی که این واحد hazard را تشخیص دهد، مقدار PC تغییر نمی‌کند، مقدار قبلی از IF به ID بافر می‌شود و واحد کنترل کار نمی‌کند.

اگر رجیستر rs یا rt در استیج ID برابر رجیستر مقصد در استیج EXE باشد، در استیج EXE هازارد داریم. همچنین اگر رجیستر rs یا rt در استیج ID برابر رجیستر مقصد در استیج MEM باشد، در استیج MEM هازارد داریم. حال اگر دستورالعمل دیکود شده برنج باشد و هازارد داشته باشیم یا سیگنال mem_to_reg در استیج EXE فعال باشد و در استیج MEM هازارد داشته باشیم، این واحد هازارد را تشخیص می‌دهد.

Datapath در صفحه بعد آورده شده است.

